(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－91297
（P2002－91297A）
(43)公開日　平成14年３月27日(2002.3.27)

(51)Int.Cl.⁷	識別記号	ＦＩ	ﾃｰﾏｺｰﾄﾞ(参考)
Ｇ０９Ｃ　1/00	６１０ ６５０	Ｇ０９Ｃ　1/00	６１０Ｂ　５Ｊ１０４ ６５０Ｚ

審査請求　未請求　請求項の数11　ＯＬ　（全 17 頁）

(21)出願番号　特願2001－207326(P2001－207326)
(22)出願日　平成13年７月９日(2001.7.9)
(31)優先権主張番号　特願2000－212813(P2000－212813)
(32)優先日　平成12年７月13日(2000.7.13)
(33)優先権主張国　日本（ＪＰ）

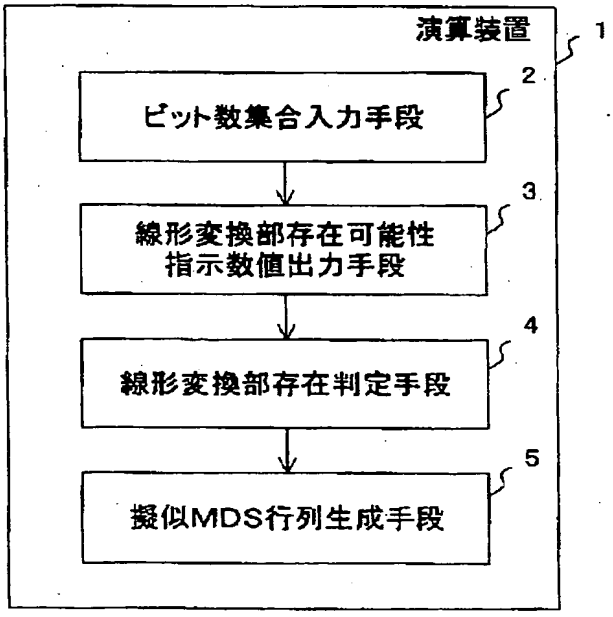
(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(72)発明者　下山　武司
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　伊藤　孝一
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(74)代理人　100074099
弁理士　大菅　義之　（外１名）

最終頁に続く

(54)【発明の名称】　Ｆ関数内部にＳＰＮ構造を用いた演算装置および演算方法

(57)【要約】
【課題】　Ｆ関数内部のＳＰＮ構造中の複数のＳボックスの間で入出力ビット数が同一でない場合に、ＳＰＮ構造内の線形変換部としてデータ拡散性能に優れたものを用いて演算を行う。
【解決手段】　演算装置１に与えられる入力データの全ビット数を非均等に分割したビット数集合Ｔの入力を受け取る手段１と、分割されたビット数をそれぞれ入・出力ビット数とする複数のＳボックスに対応する適切な線形変換部の存在可能性を示す値A_Tを出力する手段３を備える。さらにA_Tの値が正のとき適切な線形変換部が存在すると判定する手段４と、そのような線形変換部としての擬似ＭＤＳ行列を生成する手段５を備える。

本発明の原理構成ブロック図



【特許請求の範囲】

【請求項１】　複数のＳボックスと線形変換部とを備えるＳＰＮ構造をＦ関数の内部に用いた演算装置において、

該演算装置に与えられる入力データの全ビット数を非均等に分割したビット数の集合Ｔ＝｛t_1，t_2，t_3，・・・，t_r｝の入力を受け取るビット数集合入力手段と、

該分割されたビット数をそれぞれ入・出力ビット数とする複数のＳボックスに対応する適切な線形変換部の存在可能性を示す値A_Tを出力する線形変換部存在可能性指示数値出力手段とを備えることを特徴とするＦ関数内部にＳＰＮ構造を用いた演算装置。

【請求項２】　前記線形変換部存在可能性指示数値出力手段が、

前記集合Ｔの要素から任意のｋ個を選んで生成した集合の要素の和の最小値u_k（ｋ＝１，２，・・・，ｒ）を求める最小値決定手段と、

集合Ｔの要素から任意のｋ個を選んで生成した集合の要素の和の最大値v_k（ｋ＝１，２，・・・，ｒ）を求める最大値決定手段とを更に備え、

数値ｋに対して$u_k \geqq v_{k'}$（k'＝０，１，・・・，ｒ，v_0＝０）を満たすk'の最大値をｋから減算した値をw_k（ｋ＝１，２，・・・，ｒ）とし、w_kの最大値を（ｒ＋１）の値から減算して前記A_Tの値を求めることを特徴とする請求項１記載のＦ関数内部にＳＰＮ構造を用いた演算装置。

【請求項３】　前記演算装置において、

前記A_Tの値が正か正でないかを判定し、正である時前記適切な線形変換部が存在すると判定する線形変換部存在判定手段を更に備えることを特徴とする請求項１、または２記載のＦ関数内部にＳＰＮ構造を用いた演算装置。

【請求項４】　前記演算装置において、

前記線形変換部が存在すると判定された時、該線形変換部として、前記ビット数分割が均等に行われた場合のＭＤＳ行列に対応する擬似ＭＤＳ行列を生成する擬似ＭＤＳ行列生成手段を更に備えることを特徴とする請求項３記載のＦ関数内部にＳＰＮ構造を用いた演算装置。

【請求項５】　前記擬似ＭＤＳ行列生成手段が、要素が０、または１のt_i行、t_j列の部分行列M_{ij}を要素として、ｒ行、ｒ列に並べた行列をＭ＝（M_{ij}）（ｉ＝１，２，・・・，ｒ，ｊ＝１，２，・・・，ｒ）として、ｅ＝１から（A_T－１）までの各正数に対して$c(e) = e + r - A_T + 1$を求め、集合Ｔの要素をｅ個任意に選んだ集合$T_1 = \{t_{i1}, t_{i2}, \cdots t_{ie}\}$と、要素を$c(e)$個任意に選んだ集合$T_2 = \{t_{j1}, t_{j2}, \cdots, t_{jc(e)}\}$を求め、該集合（$T_1$，$T_2$）に対応する任意のあらゆるＭの小行列、および集合（T_2，T_1）に対応する任意のあらゆるＭの小行列の階数の値が、それぞれ自小行列の行数、または列数のいずれかに等しい行列Ｍを求めることを特徴とする請求項４記載のＦ関数内部にＳＰＮ構造を用いた演算装置。

【請求項６】　前記集合（T_1，T_2）に対応する小行列は、前記行列Ｍ＝（M_{ij}）を構成する前記ｒ行、ｒ列の要素としての部分行列M_{ij}のうちで、前記t_{i1}，t_{i2}，・・・，t_{ie}にそれぞれ対応する行と、t_{j1}，t_{j2}，・・・，$t_{jc(e)}$にそれぞれ対応する列とによって指定される部分行列によって構成されることを特徴とする請求項５記載のＦ関数内部にＳＰＮ構造を用いた演算装置。

【請求項７】　複数のＳボックスと線形変換部とを備えるＳＰＮ構造をＦ関数内部に用いる演算方法において、

与えられる入力データの全ビット数を非均等に分割したビット数の集合Ｔ＝｛t_1，t_2，t_3，・・・，t_r｝の入力を受け取り、

該分割されたビット数をそれぞれ入・出力ビット数とする複数のＳボックスに対応する適切な線形変換部の存在可能性を示す値A_Tを出力することを特徴とするＦ関数内部にＳＰＮ構造を用いた演算方法。

【請求項８】　前記演算方法において、

前記A_Tの値が正か正でないかを判定し、

正である時前記適切な線形変換部が存在すると判定することを特徴とする請求項７記載のＦ関数内部にＳＰＮ構造を用いた演算方法。

【請求項９】　前記線形変換部が存在すると判定された時、該線形変換部として、前記ビット数分割が均等に行われた場合のＭＤＳ行列に対応する擬似ＭＤＳ行列を生成することを特徴とする請求項８記載のＦ関数内部にＳＰＮ構造を用いた演算方法。

【請求項１０】　複数のＳボックスと線形変換部とを備えるＳＰＮ構造をＦ関数内部に用いた演算を実行する計算機によって使用される記憶媒体において、与えられる入力データの全ビット数を非均等に分割したビット数の集合Ｔ＝｛t_1，t_2，t_3，・・・，t_r｝の入力を受け取るステップと、

該分割されたビット数をそれぞれ入・出力ビット数とする複数のＳボックスに対応する適切な線形変換部の存在可能性を示す値A_Tを出力するステップとを計算機に実行させるためのプログラムを格納した計算機読出し可能可搬型記憶媒体。

【請求項１１】　複数のＳボックスと線形変換部とを備えるＳＰＮ構造をＦ関数内部に用いた演算を実行する計算機によって使用されるプログラムにおいて、

与えられる入力データの全ビット数を非均等に分割したビット数の集合Ｔ＝｛t_1，t_2，t_3，・・・，t_r｝の入力を受け取る手順と、

該分割されたビット数をそれぞれ入・出力ビット数とする複数のＳボックスに対応する適切な線形変換部の存在可能性を示す値A_Tを出力する手順とを計算機に実行さ

せるためのプログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は共通鍵ブロック暗号化方式に係り、さらに詳しくはFeistel 構造と呼ばれる構造内のＦ関数の内部で用いられる複数のＳボックスに対する入出力ビット数が、複数のＳボックスの間で同一でない場合に、複数のＳボックスの後に備えられる線形変換部として、データ拡散性のよい線形変換部を生成し、その線形変換部を用いて入力データに対する暗号化を行う暗号化装置、および暗号化方法に関する。

【０００２】

【従来の技術】高度情報化社会を迎え、情報セキュリティの確保は緊急の課題となっている。情報セキュリティの基本となるのはデータの暗号化である。高度情報化社会において高速、かつ安全な通信を実現するために、共通鍵ブロック暗号は不可欠の技術である。この共通鍵ブロック暗号のアルゴリズムとして、例えば応用分野に応じて様々な方式が提案されているが、その１つとしてFeistel 構造と呼ばれる単純な繰返し構造のアルゴリズムがある。

【０００３】図１５はこのFeistel 構造が１６段繰り返されたＤＥＳ暗号方式の説明図である。同図において入力Ｐ、例えば６４ビットは右側３２ビットと左側３２ビットとに分割され、右側３２ビットはＦ関数５１と呼ばれる非線形関数に入力され、その出力と左側３２ビットとがＸＯＲ５２によって排他的論理和がとられ、その結果は右側３２ビットとして次の段に与えられ、次の段への左側３２ビットとしては入力６４ビットのうち右側３２ビットがそのまま与えられる。

【０００４】図１６は図１５内のＦ関数５１の構成例である。入力、例えば３２ビットはビット拡張部Ｅ６１によって４８ビットに拡張され、ＸＯＲ６２によってその４８ビットと鍵K_i ４８ビットとの排他的論理和がとられ、その出力は６ビットずつに分割されてＳボックスと呼ばれる非線形関数にそれぞれ入力される。各Ｓボックス６３の出力は、例えば４ビットとされ、合計３２ビットが線形関数Ｐ６４に入力され、データの拡散が行われる。このような構造は一般にＳＰＮ（サブスティテューションパーミュテーションネットワーク）構造と呼ばれる。

【０００５】Ｓボックスは暗号装置の非線形の撹拌出力を得るため、またＳボックスに続いて行われる線形関数Ｐは、Ｓボックスによる局所的な非線形出力をデータ全体に拡散させるために用いられるが、暗号装置に組み込まれる上で拡散性のよい線形変換とは何か、具体的にどう求めるか、という研究が従来より行われている。一般に、暗号に用いられる線形変換としては、一つのＳボックスの出力が次のラウンドにおいて、出来る限り多くのＳボックスの入力に関係することが望ましいが、現在ではより拡張された線形関数として、次のような性質を満たすものがよいとされている。すなわち、Ｓボックスの入出力ビット数ｓに対して、線形変換Ｐの入力Ｘおよび出力Ｙをｓビット単位、ｔ個のブロック$X=(x_1, \cdots, x_t)$，$Y=(y_1, \cdots, y_t)$，（各x_i，y_jはｓビット）に分割した場合、

$Y=P(X)$

の入出力間で成立する任意の線形関係式

$f(x_1, \cdots, x_t, y_1, \cdots, y_t)=0$

には、入出力x_i，y_j合わせて２ｔ個の変数のうち、ｔ＋１個以上の変数が含まれている（＝係数が０ではない）というものである。

【０００６】このような性質を満たす線形変換ＰとしてＭＤＳ変換が知られている。この変換は線形変換Ｐにおけるデータの拡散性を定義する１つの概念としての分岐数を最大とする線形変換である。この分岐数は暗号に対する差分攻撃や、線形攻撃に対する強度を評価するパラメータであり、その詳細については次の文献で説明されている。

【０００７】文献）　共通鍵ブロック暗号の選択／設計／評価に関するドキュメント、通信・放送機構

図１７はそのようなＭＤＳ変換を実現する線形関数Ｐの説明図である。同図においては、４つのＳボックス７１へのそれぞれの入力、および出力は８ビットであり、合計３２ビットが入力ｘとして線形関数Ｐに与えられるものとする。線形関数Ｐへの入力ｘ、および出力ｙを、それぞれＳボックスに対応させて、８ビットずつに分割した変数x_i（ｉ＝１～４）、y_j（ｊ＝１～４）とする。

【０００８】ここでx_iに入力差分Δx_iが与えられた時、そのｉの集合を次のように書き、これを入力アクティブＳボックスと名付ける。

$\{i \mid \Delta x_i \neq 0\}$

例えばx_1，x_2に入力差分が与えられた時、この集合は｛１，２｝となる。

【０００９】この入力アクティブＳボックスに対応して、出力差分Δy_jが生じるy_jに対応して、次の集合を出力アクティブＳボックスと名付ける。

$\{j \mid \Delta y_j \neq 0\}$

これら２つの集合の和集合

$\{i \mid \Delta x_i \neq 0\} \cup \{j \mid \Delta y_j \neq 0\}$

をアクティブＳボックスと名付ける。

【００１０】そしてこの集合アクティブＳボックスの要素の数ａｃｔＳ（Ｐ）の最小値は、線形変換Ｐによって決定される。アクティブＳボックスの要素の数の最小値

$\min(actS(P))$

をアクティブＳボックスの数と呼ぶことにする。このアクティブＳボックスの数の最大値は、前述の線形関係式に含まれる変数の数（ｔ＋１）に一致するとされている。このアクティブＳボックスの数の最大値が、例えば

5となる線形変換Pが存在するとすれば、入力x_i（i＝1～4）の1個が変化すると、出力y_j（j＝1～4）の4個が変化することになり、また出力の1個は入力の5個から影響されることになる。

【0011】図18はこのようなMDS変換に相当するMDS行列の説明図である。同図においてMDS行列は、それぞれ例えば0、または1の要素からなる8行、8列の部分的な行列a_{ij}（i＝1～4，j＝1～4）から構成されている。このa_{ij}の行数と列数は、図17で説明したSボックス71の入出力ビット数に対応する。

【0012】このようなMDS行列が持つべき性質を説明する。図18の行列が、MDS行列として図17で説明した線形関数Pに望まれる高い拡散性を有するためには、部分的な行列a_{ij}を要素として考えた場合の4行、4列の全体から、行数と列数が等しい任意の小行列を選択した時に、その全ての小行列が正則であることが必要とされている。

【0013】すなわち例えば1行と1列を指定した（1，1）小行列、2行と2列を指定した（2，2）小行列、3行と3列を指定した（3，3）小行列、および行列全体と一致する（4，4）小行列の全てが、逆行列を持ち、同じ配置の行列式が0でなく、ランク（階数）がフルであるという性質を持つものとされている。

【0014】

【発明が解決しようとする課題】このように共通鍵ブロック暗号化方式におけるFeistel 構造内のF関数の中で、データの拡散に重要な役割を果たす線形変換PとしてのMDS行列の設計は、従来は複数のSボックスの入出力サイズが等しいことを前提として行われていたが、複数のSボックスの間で入出力サイズが異なる場合については、適切な線形変換Pが存在するか否か、存在するとすればその変換をどのように構成すればよいかについては、従来全く知られていないという問題点があった。

【0015】本発明の課題は、上述の問題点に鑑み、複数のSボックスの間で入出力サイズが異なる場合に、データの拡散性に優れた線形変換が存在するか否かを判定し、そのような線形変換が存在する場合に、そのような線形変換に相当する擬似MDS行列を生成し、それを用いて入力データに対応する暗号文を生成する暗号文生成装置、および生成方法を提供することである。

【0016】

【課題を解決するための手段】図1は本発明の演算装置の原理構成ブロック図である。同図は、Feistel 構造のF関数の内部に、複数のSボックスと線形変換部とを備える演算装置1の原理構成ブロック図である。

【0017】図1においてビット数集合入力手段2は、演算装置1に与えられる入力データの全ビット数を非均等に分割したビット数の集合T＝｛t_1，t_2，t_3，・・・，t_r｝の入力を受け取るものである。

【0018】また線形変換部存在可能性指示数値出力手段3は、分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応して、データ拡散性に優れた線形変換部の存在可能性を示す値、例えばアクティブSボックスの数の最大値A_Tを出力するものである。

【0019】本発明の実施の形態においては、このA_Tの値が正である時、適切な線形変換部が存在すると判定する線形変換部存在判定手段4を更に備えることも、またそのような線形変換部として、ビット数分割が均等に行われた場合のMDS行列に対応する擬似MDS行列を生成する擬似MDS行列生成手段5を更に備えることもできる。

【0020】また発明の実施の形態においては、線形変換部存在可能性指示数値出力手段3が、前述のビット数集合の要素から任意のk個を選んで生成した集合の要素の和の最小値u_k（k＝1，2，・・・，r）を求める最小値決定手段と、同様にk個を選んで生成した集合の要素の和の最大値v_kを求める最大値決定手段とを更に備え、数値kに対して$u_k \geqq v_{k'}$（k′＝0，1，・・・r，v_0＝0）を満たすk′の最大値をkから減算した値をw_kとし、w_kの最大値を（r＋1）の値から減算してA_Tの値を求めることもできる。

【0021】更に本発明の実施の形態においては、擬似MDS行列生成手段5は、要素が0、または1のt_i行、t_j列の部分行列M_{ij}を要素としてr行、r列に並べた行列をM＝（M_{ij}）（i，j＝1，2，・・・，r）とし、e－1から（A_T－1）までの各正数に対して、c（e）＝e＋r－A_T＋1求め、集合Tの要素をe個任意に選んだ集合T_1と、要素をc（e）個任意に選んだ集合T_2を求め、その集合（T_1，T_2）に対応する任意のあらゆるMの小行列、および集合（T_2，T_1）に対応する任意のあらゆるMの小行列の階数の値がそれぞれ小行列の行数、または列数のいずれかに等しい行列Mを求めることもできる。

【0022】この時、例えば集合（T_1，T_2）に対応する小行列は、前述の部分行列M_{ij}のうちで、集合T_1の各要素にそれぞれ対応する行と、集合T_2の各要素に対応する列によって指定される部分行列によって構成されることもできる。

【0023】本発明の演算方法として、複数のSボックスと線形変換部とを備えるSPN構造をF関数の内部に用いる演算方法において、与えられる入力データのビット数を非均等に分割したビット数の集合Tの入力を受け取り、分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値、例えばアクティブSボックスの数の最大値A_Tを出力する方法が用いらる。

【0024】この方法においては、発明の実施形態では、A_Tの値が正である時、適切な線形変換部が存在すると判定することもでき、またそのような線形変換部と

して、ビット数分割が均等に行われた場合のＭＤＳ行列に対応する擬似ＭＤＳ行列を生成することもできる。

【００２５】更に本発明においては、複数のＳボックスと線形変換部とを備えるＳＰＮ構造を、Ｆ関数の内部に用いた演算を実行する計算機によって使用される記憶媒体として、与えられる入力データの全ビット数を非均等に分割したビット数集合Ｔの入力を受け取るステップと、分割されたビット数をそれぞれ入・出力ビット数とする複数のＳボックスに対応する適切な線形変換部の存在可能性を示す値、例えばアクティブＳボックスの数の最大値A_Tを出力するステップとを計算機に実行させるためのプログラムを格納した計算機読出し可能可搬型記憶媒体が用いられる。

【００２６】また更に本発明においては、複数のＳボックスと線形変換部とを備えるＳＰＮ構造を、Ｆ関数の内部に用いた演算を実行する計算機によって使用されるプログラムとして、与えられる入力データの全ビット数を非均等に分割したビット数集合Ｔの入力を受け取る手順と、分割されたビット数をそれぞれ入・出力ビット数とする複数のＳボックスに対応する適切な線形変換部の存在可能性を示す値、例えばアクティブＳボックスの数の最大値A_Tを出力する手順とを計算機に実行させるためのプログラムが用いられる。

【００２７】以上説明したように、本発明によればFeistel 構造の内部のＦ関数を構成するＳＰＮ構造内で、複数のＳボックスの入出力ビット数が非均等の場合に対して、データの拡散性に優れた線形変換部の生成が可能となる。

【００２８】

【発明の実施の形態】本発明においては、Feistel 構造の内部に備えられるＦ関数を構成するＳＰＮ構造内で、複数のＳボックスの間で入出力ビット数が全て同じではない場合の暗号化アルゴリズム、およびそのアルゴリズムを用いた暗号化装置を本発明の実施形態として説明する。

【００２９】図２はそのような暗号化装置の構成ブロック図である。同図において暗号化装置は処理装置１０、入力ファイル１１、出力ファイル１２、表示装置１３、および入出力装置１４によって構成されている。

【００３０】入力ファイル１１には、例えば暗号化の対象としての平文、Feistel 構造内のＦ関数への入力データのビット数ｎ、ビット数ｎが複数のＳボックスに入力される場合の各Ｓボックスへの入力ビット数t_1，t_2，・・・t_rを要素とする集合Ｔなどが格納されている。

【００３１】処理装置１０の内部には、入力ファイル１１に格納されている集合Ｔの内容を用いて、複数のＳボックスに対するそれぞれの入出力ビット数が同じでない場合に、その複数のＳボックスの出力に対応する適切な線形変換部の存在可能性を示す数値A_Tを計算するA_T計算部１５、計算されたA_Tの値を用いてそのような線形変換部が存在するか否かを判定する線形変換部存在判定部１６、そのような線形変換部が存在すると判定された時に、そのような変換部としての擬似ＭＤＳ行列を計算する擬似ＭＤＳ行列生成部１７、生成された擬似ＭＤＳ行列を用いて、入力ファイル１１に格納されている平文に対する暗号文を生成する暗号文生成部１８などを備えている。

【００３２】出力ファイル１２には、A_T計算部１５によって計算されたA_Tの値、擬似ＭＤＳ行列、およびその擬似ＭＤＳ行列を用いた暗号化アルゴリズムなどが格納される。

【００３３】図３は本実施形態において用いられるＦ関数の内部のＳＰＮ構造の例である。入力データ３２ビットは、例えば６，５，５，５，５、および６ビットに分割され、非線形変換部としてのそれぞれのＳボックス２１に入力される。各Ｓボックスは入力ビット数と同じ出力ビット数を持ち、各Ｓボックスの出力は合成されて３２ビットとして線形変換部Ｐ２２に与えられ、その変換結果がＦ関数の出力となる。

【００３４】本実施形態においては、このように複数のＳボックスへの入出力ビット数が同じでない場合に、そのビット分割の仕方によって適切な線形変換部Ｐが存在するか否か、また存在する場合にはその線形変換部をどのように求めるかが発明のポイントとなる。

【００３５】ここで入力データのビット数ｎを非均等に分割する理由について説明する。例えば従来技術で説明した図１７では、入力３２ビットを４つに分割した８ビットずつが、４つのＳボックス７１に入力されている。このようなＳボックスは　一般的には演算の高速化のために計算機の一次キャッシュメモリにテーブルとして格納され、そのテーブルにアクセスすることによって演算が行われる。図１７ではテーブルは４つであり、４回のテーブルアクセスが必要となる。

【００３６】これに対して本実施形態では、図３に示すように例えば、入力３２ビットが６，５，５，５，５、および６ビットの６つに分割され、６個のＳボックスにそれぞれ入力される。このようにビット数の少ない６個のＳボックスに分割すると、それぞれのＳボックスに対応するテーブルのサイズは小さくなり、一次キャッシュメモリ量の少ない計算機を使用しても、演算を実行することが可能となる。

【００３７】最近の計算機の一次キャッシュメモリ量の増大の傾向につれて、１つのテーブルサイズを大きくしてテーブルアクセスの回数を減らし、演算を高速化する可能性が開かれている。そこで本実施形態においては、計算機の一次キャッシュメモリ量に対応して、ビット数分割を変更できるようなビット数分割法を用いることにする。

【００３８】前述のように３２ビットを８ビット×４に

分割している場合には、テーブルの数を３つにするためには８，１６，８ビットというような分割に変換するしか方法がなく、１６ビット入力のＳボックスに対しては2^{16}個の領域を持つテーブルが必要となってしまうことになる。それに対して図３の分割方法では、例えば２個ずつまとめて１１，１０，１１ビットの３つに分割することもでき、2^{11}個の領域を持つテーブルを計算機の一次キャッシュメモリに格納することができれば、演算の高速化が可能となる。

【００３９】図４は本実施形態における暗号文生成処理の全体フローチャートである。同図において処理が開始されると、まずステップＳ１で図２で説明した線形変換部が存在するか否かを判定するための数値A_Tが求められる。この数値A_Tとしては、前述のアクティブＳボックスの要素の数の最小値の最大値が用いられる。以後このA_Tを“アクティブＳボックス数の最大値”と呼ぶ。

【００４０】そしてステップＳ２で、求められたA_Tの値に応じて、適切な線形変換Ｐが存在するか否かが判定される。具体的にはA_Tの値が正の数である時にはそのような線形変換が存在すると判定され、０、または負の数である時にはそのような線形変換は存在しないと判定される。

【００４１】線形変換が存在すると判定されると、ステップＳ３でその線形変換を実現する行列、すなわち擬似ＭＤＳ行列が生成され、ステップＳ４でその擬似ＭＤＳ行列を用いた暗号化アルゴリズム、すなわちFeistel 構造が生成され、ステップＳ５でその暗号化アルゴリズムを用いて平文が暗号化されて、処理を終了する。

【００４２】ステップＳ２でA_Tの値が０、または負の数となり、適切な線形変換が存在しないと判定されると、ステップＳ６でエラーが発生したことを示すメッセージが出力されて、処理を終了する。

【００４３】図５は図４のステップＳ１、すなわちアクティブＳボックス数の最大値A_Tの計算処理の詳細フローチャートである。同図において処理が開始されると、まずステップＳ１０で集合Ｔの内容が入力され、ステップＳ１１で集合Ｔを構成するｒ個の要素のうちからｋ個を選んで生成された集合の要素の和の最小値u_kがｋ＝０，１，２，・・・，ｒに対して求められる。

【００４４】続いてステップＳ１２で、同様に集合Ｔの要素から任意のｋ個を選んで生成された集合の要素の和の最大値v_kが求められる。続いてステップＳ１３でｋ（＝１，２，・・・，ｒ）とｋ′（＝０，１，２，・・・，ｒ）に対して次の不等式

$u_k \geqq v_k'$ （ただし$v_0 = 0$）

を満たすｋ′の最大値をｋから減算した値がw_k（ｋ＝１，２，・・・，ｒ）として求められる。

【００４５】最後にステップＳ１４でw_kの最大値がｒ＋１から減算され、A_Tの値として求められ、処理を終了する。図６は図４のステップＳ３の処理、すなわち擬似ＭＤＳ行列生成処理の詳細フローチャートである。同図において処理が開始されると、まずステップＳ２０で分割ビット数の集合Ｔの内容に応じて、t_i行、t_j列であり、要素が０、または１となっている行列M$_{ij}$（ｉ，ｊ＝１～ｒ）を作り、そのようなｒ×ｒ個の行列M_{ij}を要素としてｒ行、ｒ列に並べた行列Ｍがランダムに新しく選択される。図３で説明したＦ関数の例ではこの行列Ｍは全体としては３２行、３２列の行列となる。ここでM_{ij}を行列Ｍの部分行列と呼ぶことにする。

【００４６】続いてステップＳ２１でｅの値が１に初期化され、ステップＳ２２でｅの値がアクティブＳボックス数の最大値A_Tから１を引いた値を越えたか否かが判定され、越えていない場合にはステップＳ２３で次式によってｃ（ｅ）の値が求められる。

【００４７】ｃ（ｅ）＝ｅ＋ｒ－A_T＋１

ステップＳ２４で集合Ｔからｅ個の要素を任意に選び、集合T_1が新しく求められ、ステップＳ２５で新しい集合T_1が選択できたか否かが判定され、選択できた場合にはステップＳ２６で同様に集合Ｔからｃ（ｅ）個の要素が任意に新しく選ばれ、集合T_2が求められ、ステップＳ２７でそのような新しい集合T_2が選択できたか否かが判定される。なおステップＳ２４、およびＳ２６で新しく選択された集合T_1、およびT_2を以下のように記述するものとする。

【００４８】$T_1 = \{ t_{i1}, t_{i2}, \cdots, t_{ie} \}$

$T_2 = \{ t_{j1}, t_{j2}, \cdots, t_{jc(e)} \}$

ステップＳ２７で新しく集合T_2が選択できたと判定されると、ステップＳ２８で行列Ｍの小行列のうちで集合T_1，T_2に対応する小行列のランク（階数）が求められる。このT_1，T_2に対応する小行列の意味については後述する。そしてステップＳ２９で求められたランクの値が　外１　または　外２　のどちら

【００４９】

【外１】

$$\sum_{p=1}^{e} t_{ip}$$

【００５０】

【外２】

$$\sum_{q=1}^{c(e)} t_{jq}$$

【００５１】か、すなわち行数と列数のいずれかに等しいか否かが判定される。等しい場合には、ステップＳ３０で行列Ｍの小行列のうちで集合T_2，T_1に対応する小行列のランクが求められ、ステップＳ３１でそのランクの値が　外３

【００５２】

【外３】

$$\sum_{p=1}^{c} t_{ip}$$

【００５３】または　外４　のいずれかに等しいか否かが判定される。

【００５４】

【外４】

$$\sum_{q=1}^{c(e)} t_{jq}$$

【００５５】ステップＳ３１でランクの値が２つの総和（行数、列数）のいずれかに等しいと判定されると、ステップＳ２６に戻りｃ（ｅ）個の要素が新たに選択され、新しい集合T_2が求められ、ステップＳ２７の判定以降の処理が繰り返される。

【００５６】そしてステップＳ２７でｃ（ｅ）個の集合T_2が新しく選択できなかったと判定されると、以前にステップＳ２４で選択された集合、すなわちｅ個の要素からなる集合T_1に対応する処理が終了したことになるため、ステップＳ２４でｅ個の要素からなる集合T_1として新しい集合が求められ、ステップＳ２５以降の処理が繰り返される。

【００５７】ステップＳ２５で新しい集合T_1が選択できなかったと判定されると、ステップＳ２１で初期化されたｅ＝１の値に対応する処理が終了したことになるので、ステップＳ３２でｅの値がインクリメントされ、ステップＳ２２以降の処理が繰り返される。

【００５８】このような処理の間に、ステップＳ２９でランクの値が２つの総和の値のいずれにも等しくないと判定された時、およびステップＳ３１で同様にランクの値が２つの総和のいずれにも等しくないと判定された時には、ステップＳ２０でランダムに選択された行列Ｍが擬似ＭＤＳ行列としては不適当なものであるものとして、ステップＳ２０で新しい行列Ｍをランダムに選択する処理以降の処理が繰り返され、ステップＳ２２でｅの値がA_T－１の値を越えたと判定されると、行列Ｍの内容が擬似ＭＤＳ行列として出力され、処理を終了する。

【００５９】図５、および図６で説明した処理について、具体例を用いて更に説明する。図３で説明した全部で３２ビットの入力ビットに対する６個のＳボックスに対応して、分割される入出力ビット数の集合は次式で与えられる。

【００６０】Ｔ＝｛６，５，５，５，５，６｝

このような集合Ｔに対応して前述の最小値u_k、および最大値v_k（v_k'）は次のようになる。

【００６１】（u_1，u_2，u_3，u_4，u_5，u_6）＝（５，１０，１５，２０，２６，３２）

（v_0，v_1，v_2，v_3，v_4，v_5，v_6）＝（０，６，１２，１７，２２，２７，３２）

その結果w_kは次式となり、その最大値は１となる。

【００６２】（w_1，w_2，w_3，w_4，w_5，w_6）＝（１，１，１，１，１，０）

最終的にアクティブＳボックス数の最大値A_Tは、このw_kの最大値を用いて次式によって求められる。

【００６３】A_T＝（６＋１）－１＝６

このA_Tの値が６、すなわち正の数であることから、このように入出力ビット数が分割された６個のＳボックスによる非線形変換に適切な線形変換が存在するということが判定される。前述のようにこの行列Ｍは全体として３２行、３２列であり、その要素が０、または１のうちからランダムに選択され、選択された行列が図６のフローチャートによって擬似ＭＤＳ行列の性質を満たすか否かが判定される。

【００６４】従ってそのような行列Ｍを生成するためには、原理的には３２行、３２列の行列の全ての要素を０、または１とした場合について図６のフローチャートの処理を繰り返して、擬似ＭＤＳ行列を求めればよいことになるが、その計算量は膨大となる。

【００６５】本実施形態では計算量を削減するための擬似ＭＤＳ行列生成法を用いることにするが、その方法については後述することとし、その方法によって求められた行列Ｍの例を図７に示す。この例の行列が図６のフローチャートの処理において、最終的にステップＳ３３で出力されるまでの過程の最初の部分について具体的に説明する。なお図７において、行列内の実線で区切られた部分は、図６のステップＳ２０で説明した行列Ｍ内の部分行列M_{ij}に相当する。

【００６６】図６に対応する処理の具体例を説明する前に、まず例えばステップＳ２８で説明したT_1，T_2に対応する小行列の意味について、図８を用いて説明する。図８において、例えば集合T_1＝｛t_2，t_3，t_6｝，T_2＝｛t_2，t_3，t_5，t_6｝とした場合には、T_1，T_2に対応する小行列として図８（ａ）に示される行列が生成され、そのランクが求められる。すなわちそれぞれが行列であるM_{ij}を部分行列とする行列Ｍから計３行と４列とが指定されて小行列が構成される。この小行列はビット単位、すなわち０、または１の要素単位では１６行、２１列の行列となる。

【００６７】また図６のステップＳ３０で説明したT_2，T_1に対応する小行列としては行として集合T_2の要素であるt_2，t_3，t_5、およびt_6に相当する行と、集合T_1の要素としてのt_2，t_3、およびt_6に対応する列が選択されて小行列が構成される。この小行列を図８（ｂ）に示す。この行列は２１行、１６列の行列である。

【００６８】ここで本実施形態におけるＭＤＳ変換としての擬似ＭＤＳ行列が持つべき性質について説明する。ｎ＝３２ビットを６個に非均等に分割した集合の例としての前述のＴに対して、アクティブＳボックスの数の最大値はA_T＝６となる。これに対してビット数の分割を

均等に行う場合にA_Tに相当する値は７であり、その差は１となる。

【００６９】前述のようにビット分割が均等な場合のＭＤＳ変換としてのＭＤＳ行列では、図８で説明したようなM_{ij}（全ての行数、および全ての列数は等しい）を要素とする行列から、任意の１行と１列を指定した（１，１）小行列、２行と２列を指定した（２，２）小行列、３行と３列を指定した（３，３）小行列、・・・を考え、そのような任意の小行列が全て正則であることがＭＤＳ行列の性質として成立する。

【００７０】これに対して擬似ＭＤＳ行列では、前述の差が１であることから、ビット分割が均等な場合に選択される小行列の行、または列のいずれかに１を加えた行列が小行列として選択され、任意の小行列のランクがフル、すなわち小行列のランクがその行数、または列数に等しくなるという性質がある。

【００７１】すなわち任意の（１，２），（２，１），（２，３），（３，２），（３，４），（４，３），（４，５），（５，４），（５，６）、および（６，５）の１０種類の小行列のランクが、それぞれの小行列の行数、または列数に等しい行列を、擬似ＭＤＳ行列として図６のフローチャートにおいて選択すべきことになる。これが本実施形態における擬似ＭＤＳ行列が持つべき性質であるが、その詳細な数学的説明（証明など）については省略する。

【００７２】ここで前述の例に戻り、図６のフローチャートに対応して、そのような性質を持つ行列Ｍの選択の過程の最初の部分の説明を続ける。まず図６のステップＳ２１でｅの値が１とされ、ステップＳ２３でｃ（ｅ）の値として２が求められる。そしてステップＳ２４で集合T_1として１個だけの要素を持つ｛t_1｝＝｛６｝が選択されたとする。またステップＳ２６でｃ（ｅ）、すなわち２個の要素を持つ集合T_2として｛t_1，t_2｝＝｛６，５｝が選択されたものとする。

【００７３】図９はこの場合にそのランクが計算されるべき、ステップＳ２８におけるT_1，T_2に対応する行列である。すなわち図８において行としては１行目、列としては１列目と２列目とが指定されることになり、小行列はM_{11}とM_{12}を成分とする行列であり、その実際の内容は図７から図９のようになる。この小行列のランクは６である。

【００７４】このランクの値、すなわち６はステップＳ２９で　外５　または　外６　のい

【００７５】

【外５】

$$\sum_{p=1}^{e} t_{ip}$$

【００７６】

【外６】

$$\sum_{q=1}^{c(e)} t_{jq}$$

【００７７】ずれかの値と等しいか否かが判定される。これらの２つの値は図９の小行列の行数と列数を示し、この場合は行数、すなわち　外７　がランクの値と等しくなり

【００７８】

【外７】

$$\sum_{p=1}^{e} t_{ip}$$

【００７９】、この小行列はフルランクであることが判定される。図１０はステップＳ３０でそのランクが計算されるべきT_2，T_1に対応する小行列の例である。前述と同様に、ここでは図８のM_{ij}のうち、行としては１行目と２行目、例としては１列目が指定されることにより、M_{11}とM_{21}とによって図１０に示す小行列が構成される。そのランクは６であり、ステップＳ３１でステップＳ２９におけると同様に２つの総和と比較され、　外８　の値と等しいこ

【００８０】

【外８】

$$\sum_{p=1}^{e} t_{ip}$$

【００８１】とが判定されて、以後の処理が続けられる。そして図６のフローチャートに従って、前述の１０種類の小行列の任意のものについて、各小行列のランクがフルであることが図７の３２行、３２列の行列に対して確認され、最終的にステップＳ３３でこの行列Ｍが擬似ＭＤＳ行列として出力されることになる。

【００８２】次に図７に示した擬似ＭＤＳ行列の生成法について説明する。この行列を生成するためには、原理的には前述のように３２行×３２列の行列の全ての要素を０、または１にランダムに変化させて、図６のフローチャートを満足する行列Ｍを探すことになるが、その計算量は膨大となる。

【００８３】そこでより能率的な方法として、本実施形態においてはまず全ビット数を３０ビットとし、３０ビットを６個に分割した集合Ｔ＝｛５，５，５，５，５，５｝に対するＭＤＳ行列を従来技術によって求め、求められた３０行、３０列の行列に対して図７に示すように最も上の行のM_{1j}（ｊ＝１～６）、最も下の行のM_{6j}（ｊ＝１～６）、最も左の列のM_{i1}（ｉ＝１～６）、および最も右の列のM_{i6}（ｉ＝１～６）に対応してそれぞれ１行、１列の要素を追加することで、擬似ＭＳＤ行列を作成することにする。

【００８４】図１１、図１２はそのような３０行、３０列のＭＳＤ行列を構成するための５行、５列の部分行列

３２個を示している。この３２個の部分行列はそれぞれ５行、５列の行列であり、各行列には０～３１の番号が付けられている。０番の行列は図１１の左上の行列であり、５行、５列の行列の要素は全て０である。５行、５列の下の“０”はこの行列に対応する（同じ配置の）行列式の値を示している。０番目の行列に対しては、当然対応する行列式の値は０である。

【００８５】例えばその下の番号１の行列に対する行列式の値は１であり、以降図１２の右下の３１番までの全ての行列に対する行列式の値も１となっている。従来技術の方法を用いることによって図１１，図１２に示した番号の５行、５列の部分行列を６行、６列に並べることによって、３０ビットを６個に均等分割した場合に対応するＭＤＳ行列の例として図１３の行列が得られる。行列内の数字は図１１，図１２で説明した各行列の番号を表わす。

【００８６】図１３に示した行列は３０行、３０列の行列であり、最も上、下の部分行列に対して１行、最も左、右の部分行列に対して１列の要素をランダムに追加し、その行列に対して図６で示したフローチャートの処理を実行することによって、図７に示した擬似ＭＤＳ行列を比較的容易に生成することができる。

【００８７】図１４は本発明を実現するためのプログラムのコンピュータへのローディングの説明図である。本発明の実施形態としての暗号化装置、例えば図２に示したシステムなどは、当然一般的なコンピュータシステムとして構成することができる。

【００８８】図１４はそのようなシステムの構成を示し、コンピュータ３１は本体３２と、メモリ３３とによって構成されている。メモリ３３はランダムアクセスメモリ（ＲＡＭ）、ハードディスク、磁気ディスクなどの記憶装置であり、本発明の特許請求の範囲第１０項のプログラムや、図４～図６で説明したプログラムなどはメモリ３３に格納され、そのプログラムが本体３２によって実行されることにより、本発明の擬似ＭＳＤ行列が求められ、入力データに対する暗号化が行われる。

【００８９】本発明を実現するためのプログラムは、プログラム提供者側からネットワーク３４を介してコンピュータ３１にロードされることも、また市販され、流通している可搬型記憶媒体３５に格納され、そのプログラムがコンピュータ３１にロードされることによって実現されることも可能である。可搬型記憶媒体３５としてはフロッピー（登録商標）ディスク、ＣＤ－ＲＯＭ、光ディスク、光磁気ディスクなど、様々な形式の記憶媒体を使用することができる。前述のプログラムなどは、このような記憶媒体に格納され、コンピュータ３１にロードされることによって、本実施形態における擬似ＭＳＤ行列が生成され、その行列を用いて入力データに対する暗号文を生成することが可能となる。

【００９０】【発明の効果】以上詳細に説明したように、本発明によれば、Ｆ関数の内部の複数のＳボックスの入出力サイズが同一でない場合において、適切な線形変換としての擬似ＳＭＤ行列の存在の有無を判定することができ、そのような行列が存在する場合にはその擬似ＭＳＤ行列を生成し、その行列を使用した暗号化を行うことによって、拡散性能に優れた暗号を生成することができ、暗号化装置の性能向上に寄与するところが大きい。

【図面の簡単な説明】

【図１】本発明の原理構成ブロック図である。

【図２】本発明の実施形態としての暗号化装置のシステム構成を示すブロック図である。

【図３】本実施形態におけるＦ関数の構成例を示す図である。

【図４】暗号文生成の全体処理フローチャートである。

【図５】アクティブＳボックスの数の最大値A_Tを求める処理の詳細フローチャートである。

【図６】擬似ＭＤＳ行列を求める処理の詳細フローチャートである。

【図７】求められた擬似ＭＤＳ行列の例を示す図である。

【図８】２つの集合に対応する小行列を説明する図である。

【図９】擬似ＭＤＳ行列の小行列の例（その１）を示す図である。

【図１０】擬似ＭＤＳ行列の小行列の例（その２）を示す図である。

【図１１】３０行×３０列のＭＤＳ行列を求めるための部分行列を示す図（その１）である。

【図１２】３０行×３０列のＭＤＳ行列を求めるための部分行列を示す図（その２）である。

【図１３】図１１，図１２の部分行列を用いたＭＤＳ行列の例を示す図である。

【図１４】本発明におけるプログラムのコンピュータへのローディングを説明する図である。

【図１５】ＤＥＳ暗号の基本構造を示す図である。

【図１６】図１５におけるＦ関数の構成例の説明図である。

【図１７】Ｆ関数内の線形変換ＰとしてのＭＤＳ変換の説明図である。

【図１８】ＭＤＳ変換としてのＭＤＳ行列の説明図である。

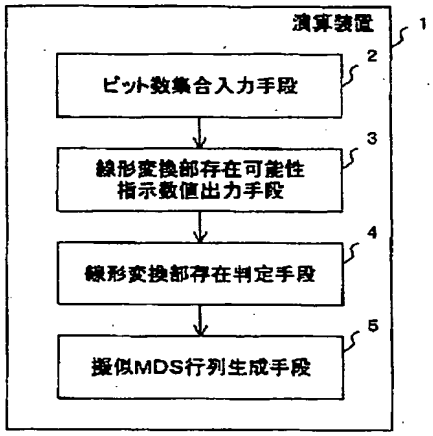
【符号の説明】

１　演算装置
２　ビット数集合入力手段
３　線形変換部存在可能性指示数値出力手段
４　線形変換部存在判定手段
５　擬似ＭＤＳ行列生成手段
１０　処理装置
１１　入力ファイル

１２　出力ファイル
１３　表示装置
１４　入出力装置
１５　A_T 計算部
１６　線形変換部存在判定部
１７　擬似ＭＤＳ行列生成部
１８　暗号文生成部

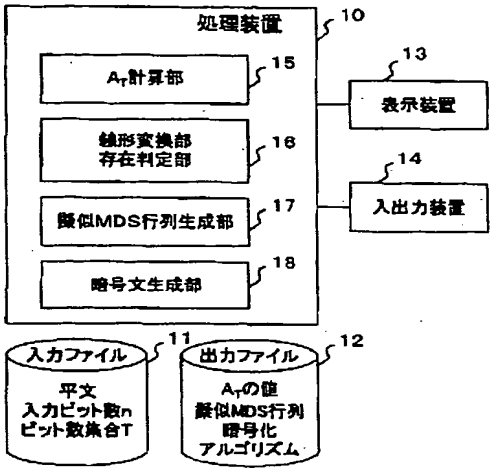
【図１】

本発明の原理構成ブロック図



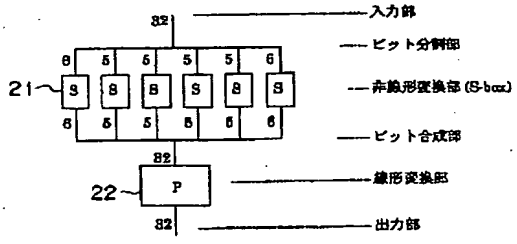
【図２】

本発明の実施形態としての
暗号化装置のシステム構成を示すブロック図



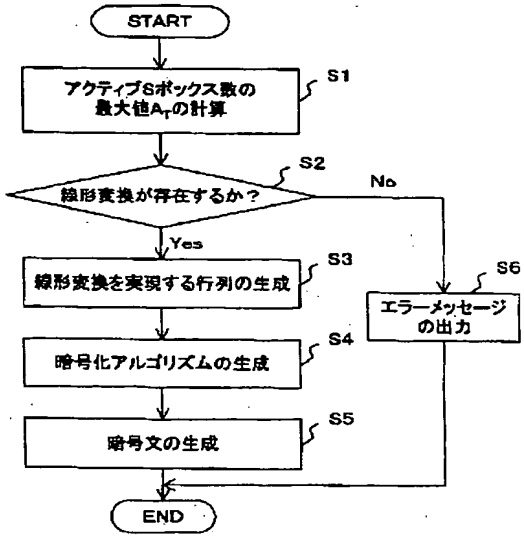
【図３】

本実施形態におけるF関数の構成例を示す図



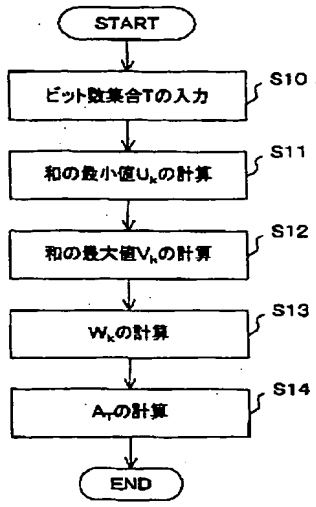
【図４】

暗号文生成の全体処理フローチャート



【図５】

アクティブSボックスの数の最大値A_Tを求める処理の詳細フローチャート



【図９】

擬似MDS行列の小行列の例（その１）を示す図

$$\left(\begin{array}{cccccc|cccccc} 1&1&1&1&1&1&1&1&0&1&1 \\ 1&1&1&0&1&1&1&0&0&1&1 \\ 0&1&0&0&1&1&0&0&0&1&1 \\ 0&0&0&0&1&1&0&0&1&1&0 \\ 0&0&0&1&1&0&0&1&1&0&0 \\ 1&1&0&1&0&0&0&0&1&1&0 \end{array}\right)$$

【図１３】

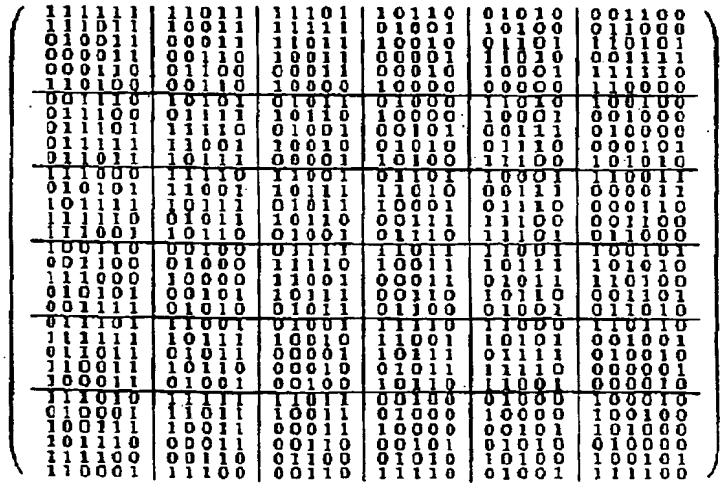
図11，図12の部分行列を用いたMDS行列の例を示す図

$$\begin{pmatrix} 31 & 27 & 29 & 22 & 10 & 12 \\ 14 & 21 & 17 & 8 & 26 & 4 \\ 24 & 30 & 25 & 13 & 17 & 19 \\ 6 & 4 & 15 & 27 & 25 & 5 \\ 29 & 25 & 9 & 30 & 24 & 22 \\ 26 & 31 & 27 & 4 & 8 & 2 \end{pmatrix}$$

【図７】

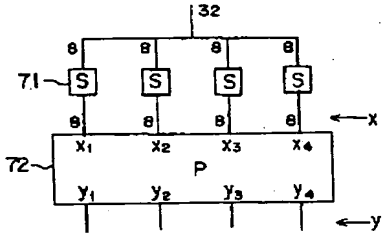
求められた擬似MDS行列の例を示す図

$M =$



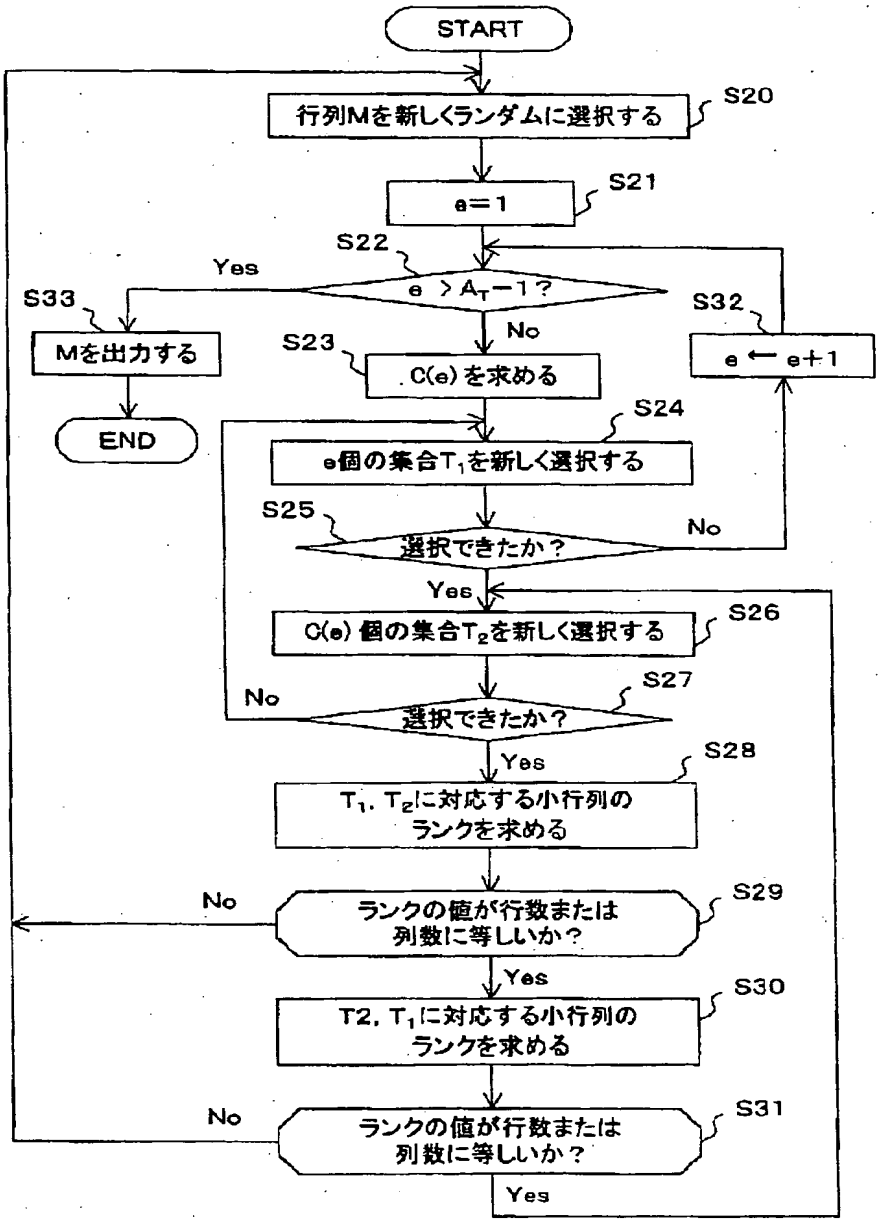
【図１７】

F関数内の線形変換PとしてのMDS変換の説明図



【図６】

擬似ＭＤＳ行列を求める処理の詳細フローチャート



【図８】

２つの集合に対応する小行列を説明する図

$$M = \begin{pmatrix} & 6 & 5 & 5 & 5 & 5 & 6 \\ 6 & M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ 5 & M_{21} & \boxed{M_{22}} & \boxed{M_{23}} & M_{24} & \boxed{M_{25}} & \boxed{M_{26}} \\ 5 & M_{31} & \boxed{M_{32}} & \boxed{M_{33}} & M_{34} & \boxed{M_{35}} & \boxed{M_{36}} \\ 5 & M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ 5 & M_{51} & M_{52} & M_{53} & M_{54} & M_{55} & M_{56} \\ 6 & M_{61} & \boxed{M_{62}} & \boxed{M_{63}} & M_{64} & \boxed{M_{65}} & \boxed{M_{66}} \end{pmatrix} \Rightarrow \begin{pmatrix} M_{22} & M_{23} & M_{25} & M_{26} \\ M_{32} & M_{33} & M_{35} & M_{36} \\ M_{62} & M_{63} & M_{65} & M_{66} \end{pmatrix}$$

(a)

$$M = \begin{pmatrix} & 6 & 5 & 5 & 5 & 5 & 6 \\ 6 & M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ 5 & M_{21} & \boxed{M_{22}} & \boxed{M_{23}} & M_{24} & M_{25} & \boxed{M_{26}} \\ 5 & M_{31} & \boxed{M_{32}} & \boxed{M_{33}} & M_{34} & M_{35} & \boxed{M_{36}} \\ 5 & M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ 5 & M_{51} & \boxed{M_{52}} & \boxed{M_{53}} & M_{54} & M_{55} & \boxed{M_{56}} \\ 6 & M_{61} & \boxed{M_{62}} & \boxed{M_{63}} & M_{64} & M_{65} & \boxed{M_{66}} \end{pmatrix} \Rightarrow \begin{pmatrix} M_{22} & M_{23} & M_{26} \\ M_{32} & M_{33} & M_{36} \\ M_{52} & M_{53} & M_{56} \\ M_{62} & M_{63} & M_{66} \end{pmatrix}$$

(b)

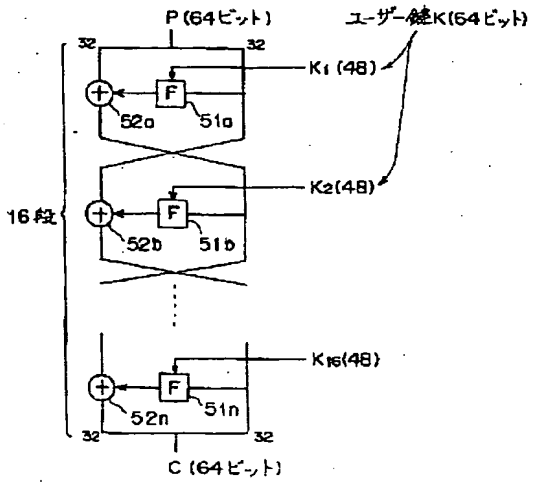
【図１０】

擬似MDS行列の
小行列の例(その2)を示す図

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

【図１５】

DES暗号の基本構造を示す図



【図１１】

30行 x 30列のMDS行列を求めるための部分行列を示す図 (その1)

```
0 : matrix[5,5]=
00000
00000
00000
00000
00000
0
1 : matrix[5,5]=
00001
00010
00100
01000
10000
1
2 : matrix[5,5]=
00010
00100
01000
10000
00101
1
3 : matrix[5,5]=
00011
00110
01100
11000
10101
1
4 : matrix[5,5]=
00100
01000
10000
00101
01010
1
5 : matrix[5,5]=
00101
01010
10100
01101
11010
1
6 : matrix[5,5]=
00110
01100
11000
10101
01111
1
7 : matrix[5,5]=
00111
01110
11100
11101
11111
1
8 : matrix[5,5]=
01000
10000
00101
01010
10100
1
9 : matrix[5,5]=
01001
10010
00001
00010
00100
1
10 : matrix[5,5]=
01010
10100
01101
11010
10001
1
11 : matrix[5,5]=
01011
10110
01001
10010
00001
1
12 : matrix[5,5]=
01100
11000
10101
01111
11110
1
13 : matrix[5,5]=
01101
11010
10001
00111
01110
1
14 : matrix[5,5]=
01110
11100
11101
11111
11011
1
15 : matrix[5,5]=
01111
11110
11001
10111
01011
1
```

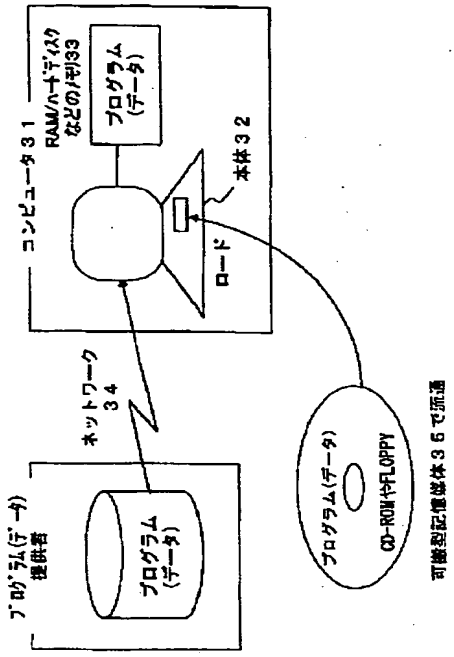
【図１２】

30行x30列のMDS行列を求めるための部分行列を示す図 (その2)

```
16 : matrix[5,5]= 24 : matrix[5,5]=
10000             11000
00101             10101
01010             01111
10100             11110
01101             11001
1                 1
17 : matrix[5,5]= 25 : matrix[5,5]=
10001             11001
00111             10111
01110             01011
11100             10110
11101             01001
1                 1
18 : matrix[5,5]= 26 : matrix[5,5]=
10010             11010
00001             10001
00010             00111
00100             01110
01000             11100
1                 1
19 : matrix[5,5]= 27 : matrix[5,5]=
10011             11011
00011             10011
00110             00011
01100             00110
11000             01100
1                 1
20 : matrix[5,5]= 28 : matrix[5,5]=
10100             11100
01101             11101
11010             11111
10001             11011
00111             10011
1                 1
21 : matrix[5,5]= 29 : matrix[5,5]=
10101             11101
01111             11111
11110             11011
11001             10011
10111             00011
1                 1
22 : matrix[5,5]= 30 : matrix[5,5]=
10110             11110
01001             11001
10010             10111
00001             01011
00010             10110
1                 1
23 : matrix[5,5]= 31 : matrix[5,5]=
10111             11111
01011             11011
10110             10011
01001             00011
10010             00110
1                 1
```

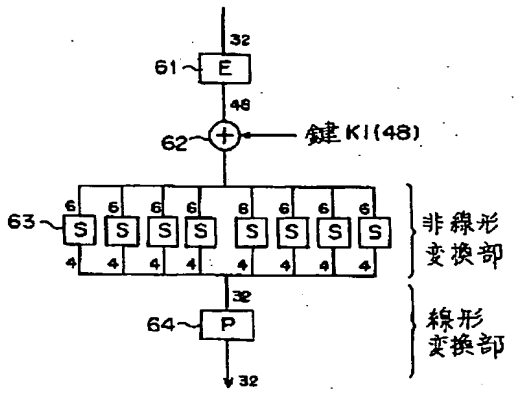
【図１４】

本発明におけるプログラムの
コンピュータへのローディングを説明する図



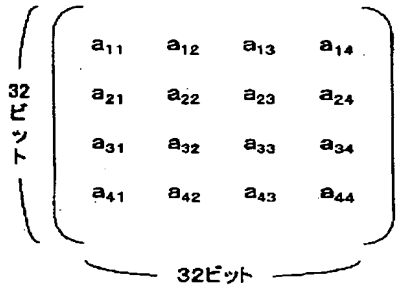
【図１６】

図15における F 関数の構成例の説明図



【図１８】

MDS変換としてのMDS行列の説明図



フロントページの続き

(72)発明者　武仲　正彦
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　鳥居　直哉
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　矢嶋　純
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　屋並　仁史
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

(72)発明者 横山 和弘
神奈川県川崎市中原区上小田中４丁目１番
１号 富士通株式会社内

Ｆターム(参考) 5J104 AA19 JA13 JA17 NA02